第26卷 第6期　　工 程 数 学 学 报　　Vol. 26 No. 6
2009年12月　　CHINESE JOURNAL OF ENGINEERING MATHEMATICS　　Dec. 2009

文章编号:1005-3085(2009)06-0977-08

基于环分布的频繁子图挖掘算法*

董安国[1,2]， 高　琳[2]， 邱在秦[3]， 常安定[1]， 赵建邦[2]

(1- 长安大学理学院，西安 710064； 2- 西安电子科技大学计算机学院，西安 710071；
3- 西安石油大学理学院，西安 710065)

摘　要: 频繁子图挖掘主要涉及到子图搜索和子图同构问题。对子图搜索问题，本文提出了环分布的概念，并构造了基于环分布的子图搜索算法；对子图同构问题，本文利用度序列和特征值构造了两种算法，分别用于对有向图和无向图的同构判别。利用同构算法对搜索出的子图进行同构分类，根据分类结果得到频繁子图。实验结果表明，本算法的效率优于现有算法。

关键词: 频繁子图；子图搜索；子图同构；特征值；度序列

分类号: **AMS**(2000) 42C40; 65T60　　**中图分类号**: TP311.12; Q811.4　　**文献标识码**: A

1　引言

在生物信息学、社会网络、集成电路的布局布线、Web数据挖掘、网络工程等众多领域都积累了大量的关于图的数据，这些数据中包含了大量重要的信息，而频繁发生的子图往往是这些信息的载体。所以，频繁子图的挖掘问题具有重要的应用价值。子图挖掘的研究工作始于1994年Holder等人[1]提出的著名子图挖掘算法SUBDUE；1994年，Yoshida等人[2]提出了一个子图挖掘算法GBI，它类似SUBDUE算法，但采用了不同的启发式搜索策略。2000年，Inokuchi等人[3]提出了一个基于Apriori思想的频繁子图模式挖掘算法AGM，随后在AGM算法的基础上，提出了挖掘连通频繁子图的算法AcGM[4]。2002年以后，各种不同的子图挖掘的算法被提出来，比较有影响的算法有，Yan and Han[5]提出的gSpan算法，Kuramochi等人[6]提出的FSG算法，Wernicke[7]提出的ESU算法。除了上述这些图模式挖掘的通用算法外，研究人员还提出了大量运用于实际问题的图模式挖掘算法[8,9]。综上所述，图模式挖掘算法多种多样，应用广泛，但由于问题本身的复杂度导致现有算法的效率还不尽如人意。

子图同构是频繁子图挖掘中的一个关键步骤。在一般意义下，图的同构是NP-完全问题[10]，有人试图用图的一组不变量来确定图的同构，如回路数、树数、连通片数等，这些尝试都归于失败，因为不同构的图也会出现完全相同的不变量[11]。所以子图同构问题成为子图挖掘的一个瓶颈。目前，子图同构的最常用的是最小编码算法[12]，对无标签图特别是无向图，这种算法效率不高。本文通过引入综合度，提出局部序号置换算法；对无向图，利用特征值理论，构造了特征值同构算法。实验表明，在频繁子图挖掘的两个环节，本文的算法均优于文献[7]的算法。

本文研究的主要内容是:

收稿日期: 2007-12-20.　**作者简介**: 董安国(1964年9月生)，男，硕士，教授. 研究方向：计算生物学.
***基金项目**: 国家自然科学基金(60574039)；陕西省自然科学计划项目(SJ08-ZT15)；长安大学科技发展基金(07J04).

1) 提出了一种基于节点环分布的子图搜索算法ESR (Enumerate Subgraphs based on Ring)；

2) 构造了子图同构算法：局部序号置换算法和特征值同构算法；

3) 对5个真实生物网络进行了仿真试验研究，找出了频繁子图，并对算法的效率进行了比较分析。

2 相关定义及其基本结论

定义1 $G=(V,E)$为一个给定的图，称$G_s=(V_s,E_s)$为G的子图，当且仅当$V_s\subseteq V$且$E_s\subseteq E$。特别地，如果E_s包含E中所有连接V_s中节点的边，则称$G_s=(V_s,E_s)$为G的导出子图。

特别指出，本文所要搜索的子图是指连通的导出子图，并记k阶子图为G^k。

定义2 设G为一个m阶(有m个节点)图，称复数D_i为节点i的综合度，其中D_i的实部为节点i的出度SO_i，虚部为节点i的入度SI_i；对无向图，综合度即为节点的度；规定复数的序为实部和虚部的字典序，将D_i $(i=1,2,\cdots,m)$按升序排列得到的序列称为顺序度向量，并记为T，即$T=(D_{(1)},D_{(2)},\cdots,D_m)$，其中$D_{(i)}$表示$D_i$ $(i=1,2,\cdots,m)$按升序排列后的第i个元素。

定义3 对相同阶数的所有连通图，按同构关系将其分成若干个类，称这样的类为等价类。

定义4 对单位矩阵作两行互换所得的矩阵称为初等置换矩阵，对单位矩阵进行若干次两行互换所得的矩阵称为置换矩阵。

定义5 若存在一个置换矩阵P，使$A_1=P^{-1}A_2P$，则称A_1, A_2是置换相似。

由图的同构定义以及矩阵特征值理论易得以下结论。

定理1 设F为初等置换矩阵，则$F^{-1}=F$，且置换矩阵可以表示成一系列初等置换矩阵的乘积。

定理2 A_1, A_2分别表示图G_1, G_2的连接矩阵，则G_1和G_2同构的充分必要条件是存在置换矩阵P使$A_1=P^{-1}A_2P$。

推论1 G_1和G_2同构的充分必要条件是A_1, A_2置换相似。

定理3 如果G_1和G_2同构，则它们的顺序度向量相等，其连接矩阵A_1, A_2有相同的特征值。

定义6 在图$G=(V,E)$中，从节点i到j需要经历的最少边数称为节点i到节点j的距离。

定义7 设v是图G是一个节点，与v的距离为l的节点的集合称为节点v的第l个环。设G^k是一个k阶连通图，v_0是G^k的一个节点，则v_0最多有$k-1$个环，其各环中分布的节点个数$(x_1,x_2,\cdots,x_{k-1})$称为G^k以v_0为中心的环分布，所有可能的环分布用$P^{(k)}$表示，即$P^{(k)}=(P_{ij}^{(k)})_{N\times(k-1)}$，$P_{ij}^{(k)}$表示第$i$类环分布在节点$v_0$的第$j$个环上的节点个数，$N$表示环分布类型数。例如，3阶连通图所有可能的分布类型有两种，即

$$P^{(3)}=\begin{bmatrix}2&1\\0&1\end{bmatrix}.$$

一般地，$P^{(k)}$的确定可由如下的定理4给出。

定理4 设$P_1^{(k)}$表示将$P^{(k)}$的第一列元素加1，并在最后补上一列0得到的矩阵，$P_2^{(k)}$表示在$P^{(k)}$的第一列前面补上一列1得到的矩阵，则$P^{(k+1)}$就是$P_1^{(k)}$和$P_2^{(k)}$上下拼接得到的矩

阵，即

$$P^{(k+1)} = \begin{pmatrix} P_1^{(k)} \\ P_2^{(k)} \end{pmatrix}.$$

证明　由环分布的定义，结论显然。

由定理4可知，k阶连通图共有2^{k-2}个不同的环分布类。

3　基于环分布的子图搜索算法

在搜索n阶图G的所有k阶子图S_k时，先从G中取一个节点，不妨设v_1，搜索所有包含节点v_1的k阶子图，记为$S_k(v_1)$，然后从G中删除v_1以及与之相连的边，再从G中取一个节点v_2，重复上述过程，得到$S_k(v_2), S_k(v_3), \cdots, S_k(v_{n-k+1})$，显然，$S_k(v_i) \cap S_k(v_j) = \phi$，$i \neq j$，且

$$S_k = \bigcup_{i=1}^{n-k+1} S_k(v_i).$$

在搜索$S_k(v_1)$时，由于$S_k(v_1)$中的每个子图都包含节点v_1，这些子图相对于v_1的所有可能的环分布有2^{k-2}种，依次搜索以v_1为中心的各种环分布的子图，记第i种环分布的子图为$S_k^i(v_1)$，则$S_k^i(v_1) \cap S_k^j(v_1) = \phi$，$i \neq j$，且

$$S_k(v_1) = \bigcup_{i=1}^{2^{k-2}} S_k^i(v_1).$$

由于$S_k^i(v_i)$是指以v_1为中心的第i种环分布的子图，所以其余的$k-1$个节点在v_1的各个环上，分布的节点数为$P_{ij}^{(k)}$，$j = 1, 2, \cdots, k-1$。搜索时，从v_1开始在它的各个环上扩张，直到k个节点。记当前搜索的子图为G_s，则给G_s赋初值v_1，即$G_s = \{v_1\}$，计算v_1的第一个环R_1(与v_1有边相连的节点集)，如果R_1中节点数不小于$P_{i1}^{(k)}$，则在R_1中遍历$P_{i1}^{(k)}$个节点的选择(在R_1中任选$P_{i1}^{(k)}$个节点)，对每一个选择w_1(w_1是R_1中$P_{i1}^{(k)}$个节点组成的节点集)，计算与w_1至少一个节点有连边，而与G_s无连边的节点集R_2，再将w_1加到G_s中，即$G_s = G_s \cup w_1$。如果R_2中节点数不小于$P_{i2}^{(k)}$，则在R_2中遍历$P_{i2}^{(k)}$个节点的选择，对每一个选择w_2，计算与w_2至少一个节点有连边，而与G_s无连边的节点集R_3，更新$G_s = G_s \cup w_2$，依次类推，直到G_s中的节点数等于k。根据上述的算法描述，$S_k^i(v_1)$中的子图不会被遗漏，所得不同的G_s相对于v_1的环数是一样的，每个环上分布的节点数是相同的，但不同的G_s至少有一个环上的节点是不同的，所以不会重复搜索到同一个子图。

上述算法形式化描述如下：

输入：图G的连接矩阵A，子图阶数k；输出：图G的所有k阶子图subgraph。

```
for=1 to n-k+1 //对所有的节点
    subgraph=1
    w=1
    R=与节点w相连的节点
    for i=1 to 2^(k-2) //对所有的环
        s=1
        subgraph=EXTENDSUBGRAPH(subgraph, i, s, R)
    endfor
```

```
        A← 删除 A 的第一行和第一列
endfor
```

函数 subgraph=EXTENDSUBGRAPH(subgraph, i, s, R) 是第 i 个分布类在第 s 个环中扩展节点 (每步扩展 P_{is} 个节点)

```
Function subgraph=EXTENDSUBGRAPH(subgraph, i, s, R)
if |subgraph| == k //k 为子图阶数
        输出 subgraph //
        return
endif
while |subgraph| ≠ 0 and |R| ≥ P_is
        for all w //w 表示从 R 中取的 P_is 个节点的一个选择
                R1=neighbor(A, subgraph, w)
                subgraph= subgraph ∪ w
                s = s + 1
                subgraph=EXTENDSUBGRAPH(subgraph, i, s, R1)
        endfor
endwhile
return
```

函数 R=neighbor (A, subgraph, w) 是计算与 w 至少有一个节点有连边而与 subgraph 没有连边的节点，即 subgraph 的下一个环。

4 同构分类算法

在频繁子图的搜索过程中，每搜索到一个子图都需要判定是否与已有的某子图相同 (同构)，然后才能够确定这个子图的类别并进行频数的统计。在一般意义下，图的同构是 NP-完全问题[10]，尽管子图阶数不高，但由于搜索到的子图数量庞大 (见实验结果数据)，同构分类的运算量很大，下文将以代数理论为基础，利用度序列和特征值构造了两种子图同构算法，分别用于对有向图和无向图的同构判别。

4.1 有向图的同构算法

根据以上定义，两个具有 k 个节点的同构图，其连接矩阵未必相同，但可以改变其中一个图节点的排列次序 (不改变拓扑结构)，使它们的连接矩阵相等，然而节点的排序共有 $k!$ 种，所以直接利用定义来判断图的同构运算量很大，本节将通过引入综合度，根据综合度是否相等将 k 个节点分成 j 组，第 i 组为 $t_i\,(i = 1,2,\cdots,j)$ 个节点，这样就将节点排序数从 $k!$ 降到 $\prod\limits_{i=1}^{j} t_i!$。

设 G 为一个 k 阶有向图，计算其顺序综合度向量；根据定理3，如果 G_1 和 G_2 的顺序度向量不同，G_1 和 G_2 一定不同构，否则，需要做进一步的置换比较才能确定它们是否同构。

算法思想描述如下：

步骤 1：输入 G_1 和 G_2 的连接矩阵 A_1 和 A_2；

步骤 2：计算 G_1 和 G_2 综合度向量和顺序综合度向量；

如果顺序综合度向量不同，则 G_1 和 G_2 不同构；

如果顺序综合度向量相同，G_1 的各节点按综合度从小到大重新标号，如果有若干个节

点综合度相同，则这些节点之间排序可任取一种，按新的节点标号更新连接矩阵A_1, G_2的各节点按综合度从小到大重新标号，如果若干个节点综合度相同，则对综合度相同的节点列出所有可能的排序，设共有m种(每个节点综合度都不同则$m=1$)，按各种可能的节点排序更新矩阵A_2产生m个连接矩阵$A_{21}, A_{22}, \cdots, A_{2m}$，如果存在$i\,(1 \leq i \leq m)$使$A_1 = A_{2i}$则$G_1$和$G_2$同构，否则不同构。

该算法对无向图的效率低于有向图，针对算法的实验也说明了这一点。为此，下文将针对无向图设计了特征值同构算法。

4.2 无向图的同构算法

$G^{(k)}$表示k阶无向连通图等价类的集合，由于每一个等价类中的图拓扑结构都一样，是同一个图，故设第i个等价类的图为$G_i^{(k)}\,(i=1,2,\cdots,N(k))$，其中$N(k)$表示$k$阶无向连通图等价类的个数，即$G^{(k)} = \{G_i^{(k)} \mid (i=1,2,\cdots,N(k))\}$，设$T^{(k+1)}$为$G^{(k)}$中的每一个图增加一个节点所得到的$k+1$阶连通图的全体，$EG^{(k+1)}$表示$T^{(k+1)}$中图的等价类全体。

定理5 对$G_i^{(k)}\,(i=1,2,\cdots,N(k))$中的每一个图增加一个节点所得到的所有$k+1$阶连通图中等价类的个数等于$k+1$阶连通图的等价类个数，即$|EG^{(k+1)}| = |G^{(k+1)}|$。

证明 如果存在一个连通图$G \in G^{(k+1)}$，则一定存在一个节点v，使$G' = \{G \setminus v\} \in G^{(k)}$，所以$G \in T^{(k+1)}$，从而$G \in EG^{(k+1)}$，即$G^{(k+1)} \subset EG^{(k+1)}$；显然$EG^{(k+1)} \subset G^{(k+1)}$，所以$|EG^{(k+1)}| = |G^{(k+1)}|$。

定理5表明，要得到$k+1$阶无向连通图的所有等价类，不需要对所有$k+1$阶无向连通图进行同构分类，只需要对$T^{(k+1)}$中的图进行同构分类。以定理5为理论依据，可以利用计算机证明出图同构的一个充分必要条件，并用这一条件来判别图的同构。

定理6 设G_1, G_2为节点数不大于7的无向连通图，G_1和G_2同构的充分必要条件是A_1和A_2有相同的特征值。

本定理的证明由计算机通过计算来完成，其基本思想是：给定子图阶数k，根据定理5的结论和递推关系产生一些列矩阵(这些矩阵对应的子图所包含的类别和全部k阶子图包含的类别一样)，分别利用4.1的算法和特征值进行分类，如果分出的类别一样，则表明对k阶子图，两个矩阵特征值一样可以作为对应子图同构的等价条件，将k加1重复上述过程；如果分出的类别不一样，表明特征值不能作为子图同构的等价条件。经计算，当$k=8$时分的类不一样，这样就证明了定理6。

设Q表示由$k-1$阶各等价类图增加一个节点得到的所有k阶连通图，$LQ1$表示Q中全部等价类(每一类中的图只用一个图来代表)，由定理5，$LQ1$等于全部k阶连通图产生的等价类；$T=1$表示定理6成立。

计算机证明程序流程如下：

步骤1:　Input $k=2$；$T=1$ 表示2阶图定理6成立

步骤2:　Input A=[0,1;1,0]；Q=A，LQ1=A 输入2阶连通图的类(只有一类)

步骤3:　While $T=1$

步骤4:　　Q=EXTEND(LQ1，k) //在LQ1上扩展一个节点产生$k+1$阶图

步骤5:　　对Q中的图利用2.1的算法进行分类，LQ1表示所得的类别

步骤6:　　对Q中的图按特征值是否相等进行分类，LQ2表示所得的类别

步骤7:　　If LQ1=LQ2 (分类结果一致，说明特征值相同可以作为同构的充分必要条件)

步骤8:　　　$T=1$；$k=k+1$

步骤9:　　Else (表明一定存在两个不同构的图，其连接矩阵的特征值相同)

步骤10:　　　$T=0$

步骤11: Endif
步骤12: Endwhile
步骤13: Output T, k
程序执行结果如下：

当节点数$k=8$，程序结束，得到两个8节点的不同构图有相同的特征值，如图1和图2所示，它们不同构，但其连接矩阵的特征值均为(-2.0000,-1.6624,-1.0000,-0.7574,0.4249,1.0000,1.4959,2.4989)。但当$k<8$时，定理6成立。

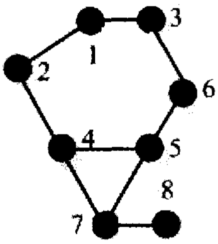


图1: 8阶无向图a

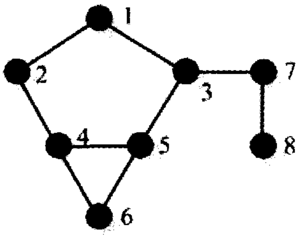


图2: 8阶无向图b

根据定理6，对低于8阶的无向子图进行同构分类时，可以利用连接矩阵的特征值进行判断，而在频繁子图挖掘问题中，子图的阶数小于8[13]，所以特征值同构算法在无向网络的频繁子图挖掘中可以作为一般性的结论。

5 仿真试验研究

实验的软件是：Matlab7.1；实验的数据是基因调控网络E. coli和Yeast，Electronic，SeaUrchin和蛋白质相互作用网络Protein，其中Protein是无向网络，数据来源是文献[14]。

5.1 子图搜索速度比较

对本文及文献[7]的子图搜索算法分别进行编程计算，对5个真实的网络，分别搜索了3-7阶子图，并统计了搜索时间，具体结果见表1(3阶子图的搜索结果未在表中列出)，从表中可以看出，随着子图节点的增加，其数量急剧上升，搜索时间也增加，但本文的搜索算法(ESR)在单位时间内搜索到的子图数量基本不变，所以本算法的效率高于文献[7]。

表1: 子图数量及搜索时间表

	边/节点		4阶子图		5阶子图		6阶子图		7阶子图	
			子图数量	时间	子图数量	时间	子图数量	时间	子图数量	时间
E.coli	519/423	ESR	83893	7.3	1433502	44	22532584	559.9	319521581	9199
		ESU	83893	9	1433502	170	22532584	>2h	319521581	>3h
Electronic	819/512	ESR	10168	8.0	53155	12	303689	40	1781484	158
		ESU	10168	2	53155	13	303689	70	1781484	>3h
SeaUrchin	81/45	ESR	2212	0.5	11043	0.9	49320	4	196082	13
		ESU	2212	1	11043	1	49320	1	196082	14
Yeast	1079/688	ESR	183174	17	2508149	125	32883898	1501	416284878	20561
		ESU	183174	30	2508149	600	32883898		416284878	
Protein	716/270	ESR	118129	5.4	1685010	40	22990600	493.6	297549099	8268
		ESU	118129	8	1685010	180	22990600		297549099	

说明：ESR是本文算法，ESU是文献[7]的算法，时间单位是秒；表中空缺部分表示作者按文献[7]编写的程序在2小时内未算出结果，对应的子图数量来自参考文献[7]。

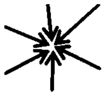
5.2 子图同构分类

由表2可见，不同的网络，搜索到的子图数量差别很大，同一个网络，随着子图阶数的提高，其数量急剧增加。要搜索阶数较高的频繁子图，同构分类的运算量很大，所以当子图数量很大时，从搜索到的子图集中随机抽取一定数量的样本(本文抽取100万个，子图数量小于100万就精确计算频率)来估计子图频率。表3列出了几种同构算法的运行时间，表4列出了E.coli和Protein网络的7阶频繁子图(前3个)。

表2: 同构分类时间比较

	边数	节点数	分类时间		
			特征值算法	局部置换算法	文献[5]算法
E.coli (有向图)	519	423	无效	1617	2285
Protein (无向图)	716	270	535	3268	4162

表3: 前三个7阶频繁子图

序号	E.coli		Protein	
	频繁子图	频 率	频繁子图	频 率
1		0.4473		0.0784
2		0.1728		0.0718
3		0.0856		0.0513

实验结果分析：1) 同构分类的时间代价较高，主要是因为在分类过程中，每一个子图都要与已有的类进行比较，如果它与已有的某个类同构，则将该类的计数加1，否则，将该子图归到一个新的类。所以，对每一个子图的分类需要进行多次的同构比较，运算量非常大，从而同构分类问题成为频繁子图挖掘中的一个瓶颈问题。2) 在对无向图的同构分类中，本文提出的特征值算法效率明显高于局部置换算法和文献[10]的算法。3) 顺序度算法的效率高于文献[10]的最小编码算法，顺序度算法和最小编码算法对无向图效率低。

6 总结与展望

本文首先提出了基于环分布的子图搜索算法，它包括环分布类型的确定和子图搜索；为了得到子图的频率，必须要涉及子图同构算法；为此又构造了综合度同构算法和特征值同构算法。通过对5个真实网络数据的仿真实验研究，表明本文提出的算法比现有算法的效率高。在

大型网络 (生物网络，社会网络等) 的模体发现问题中，如果能够知道模体的一些先验信息，就可以在搜索过程中排除一些结构类型，进一步提高搜索效率。由于本文的算法效率高，我们下一步的研究工作是将该算法应用到生物网络数据中，通过对真实网络数据及随机网络的频繁子图的比较找出生物模体单元。

参考文献：

[1] Holder L B, Cook D J, Djoko S. Substructure discovery in the subdue system[C]// Proceedings of ACM SIGKDD, the 1994 International Conference on Knowledge Discovery in Database(KDD'94), July 1994: 169-180

[2] Yoshida K, Motoda H, Indurkhya N. Graph-based induction as a unified learning framework[J]. Journal of Applied Intelligence, 1994, 4: 297-328

[3] Inokuchi A, Washin T, Motoda H. An apriori-based algorithm for mining frequent substructures from graph data[C]// Proceedings of the 2000 Europe Conference on Principle of Data Mining and Knowledge Discovery(PKDD'00), Lyon, France, September, 2000: 13-23

[4] Inokuchi A, Washin T, Nishimura K, *et al.* A fast algorithm for mining frequent connected subgraphs[R]. Research Report RT-0448, IBM Tokyo Research Lab, 2002

[5] Yan X, Han J. gSpan: Graph-based substructure patterns mining[C]// Proceedings of IEEE the 2002 International Conference on Data Mining(ICDM'02), Maebashi, Japan, December, 2002: 721-724

[6] Michihiro Kuramochi, George Karypis. An efficient algorithm for discovering frequent subgraphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1038-1051

[7] Wernicke S. Efficient detection of network motifs[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2006, 3(4): 347-359

[8] Kashtan N, Itzkovitz S, Milo R, *et al.* Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs[J]. Bioinformatics, 2004, 20(11): 1746-1758

[9] Hu H, Yan X, Huang Y, *et al.* Mining coherent dense subgraphs across massive biological networks for functional discovery[J]. Bioinformatics, 2005, 21(1): 213-221

[10] Toran J. On the hardness of graph isomorphism[J]. SIAM Journal on Computing, 2004, 33(5): 1093-1108

[11] 李锋，陆韬. 任意图同构判定及其应用[J]. 复旦学报，2006, 45(4): 480-484

[12] McKay B D. Practical graph isomorphism[J]. Congr Numer, 1981, 30: 45-87

[13] Mason O, Verwoerd M. Graph theory and networks in biology[EB/OD]. http://www.hamilton.ie/systemsbiology/ files/2006/graph-theory-and-networks-in-biology.pdf

[14] www.weizmann.ac.il/mcb/UriAlon/groupNetworksData.html[EB/OD]. http://dip.doe-mbi.uda.edu

A Ring Distribution Based Algorithm for Finding Frequent Subgraphs

DONG An-guo[1,2], GAO Lin[2], QIU Zai-qin[3], CHANG An-ding[1], ZHAO Jian-bang[2]

(1- School of Science, ChangAn University, Xi'an 710064;
2- School of Computer Science and Technology, Xidian University, Xi'an 710071;
3- School of Science, Xi'an Shiyou University, Xi'an 710065)

Abstract: Frequent subgraph mining includes subgraph search and isomorphism problems. For the subgraph searching, we propose the definition of a ring distribution and provide a novel subgraph search algorithm based on the ring distribution. Furthermore, by using the degree sequence and eigenvalue, we present two algorithms for subgraph isomorphism in directed and undirected graphs, respectively. Finally, we experimentally evaluate the performance of our algorithms by using real networks. The simulation results show that our algorithm is more effective than existing algorithms.

Keywords: frequent subgraph; subgraph search; subgraph isomorphism; eigenvalue; degree sequence